1. Relational schema for Library

a)

```
SELECT DISTINCT LastName
FROM Reader
WHERE Address = ‘Moscow’;
```

b)

```
SELECT Author, Title
FROM Book, Publisher
WHERE PubKind = ‘Science’ OR PubKind = ‘Reference’;
```

c)

```
SELECT DISTINCT Author, Title
FROM Reader, Borrowing, Book
WHERE FirstName = ‘Ivan’ AND LastName = ‘Ivanov’;
```

d)

```
SELECT ISBN
FROM BookCat
WHERE CategoryName = 'mountains' AND
             ISBN NOT IN
            (SELECT ISBN
            FROM BookCat
            WHERE CategoryName = 'travel');
```

e)

```
SELECT DISTINCT LastName, FirstName
FROM Reader, Borrowing
WHERE ReturnDate NOT NULL;
```

f)

```
SELECT DISTINCT LastName, FirstName
FROM Reader, Borrowing
WHERE FirstName != ‘Ivan’ AND LastName != ‘Ivanov’ AND ISBN IN
```

```
            (SELECT ISBN
            FROM Reader, Borrowing
            WHERE FirstName = ‘Ivan’ and LastName = ‘Ivanov’);
```

2. Relational schema for Trains

a)

```
SELECT DISTINCT fromstation fst, tostation
 FROM connections con1
 WHERE tostation NOT IN
    (SELECT tostation
    FROM connections
    WHERE fromstation IN
        (SELECT tostation
        FROM connections con2
        WHERE fromstation=fst AND con1.trainnr = con2.trainnr AND con1.departure <= con2.departure AND con1.arrival >= con2.arrival
        ));
```

3. Represent outer join as relational algebra expression using only basic operations (select, project, cartesian, rename, union, minus)

$R = (A_1, \ldots, A_n, B_1, \ldots, B_k)$

$S = (B_1, \ldots, B_k, C_1, \ldots, C_m)$

$$\Pi_{A_1,\ldots,A_n,R.B_1,\ldots,R.B_k,C_1,\ldots C_m}(\sigma_{R.B_1=S.B_1 \wedge \ldots \wedge R.B_k=S.B_k}(R \times S))$$
$$\cup \Pi_{A_1,\ldots,A_n,NULL_1,\ldots NULL_n}(\sigma_{A_1,\ldots,A_n}(R))$$
$$\cup \Pi_{NULL_1,\ldots,NULL_m,\, C_1,\ldots C_m}(\sigma_{C_1,\ldots,C_m}(S))$$